

# 모바일 음성 웹 페이지의 자동 생성 시스템에 관한 연구

고유정, 김윤중  
한밭대학교 컴퓨터공학과  
e-mail : youlony@hanbat.ac.kr

## A Study On the Automatic Generation System of Mobile Voice Web Page

You-Jung Ko, Yoon-Joong Kim  
Dept. of Computer Engineering, Hanbat National University

### 요 약

모바일 기기는 화면의 크기가 작아 스타일러스나 펜으로 웹 콘텐츠를 이용하기에는 불편함이 있다. 이에 따라 음성으로 웹의 콘텐츠를 개발하기 위한 표준 언어인 VoiceXML(Voice Extensible Markup Language), SALT(Speech Application Language Tags)가 빠르게 보급되고 있다. 이를 이용하기 위해서는 기존의 모바일 웹페이지를 음성 웹 표준 기술에 맞게 변환해줘야 한다. 따라서 본 논문에서는 WML(Wireless Markup Language)로 구성된 모바일 웹 페이지를 SALT 음성기술을 이용하여 음성명령이 가능한 모바일 음성 웹 페이지(WML+SALT)로 자동 생성하는 시스템을 구현 하고자 한다. 이에 따라 사용자는 음성명령을 통해 콘텐츠를 제어함으로써 편리함을 제공하고, 개발자는 자동 생성 시스템을 이용함으로써 기존의 모바일 웹 페이지를 음성 웹 페이지를 변환하기 위한 개발시간과 비용을 감소할 수 있다.

### 1. 서론

최근 무선 데이터통신 기술의 발달 및 핸드폰, PDA와 같은 휴대용 무선 장비 수요의 증가에 따라 휴대용 무선 장비를 통해 웹상의 정보를 이용하고자 하는 요구가 점점 늘어나고 있으며, 휴대용 무선 장비를 통한 웹상의 정보 접근이 점차 현실화되고 있다.[1] 하지만 작은 화면에 스타일러스나 펜을 입력하여 웹 콘텐츠를 선택시 불편함이 따른다. 이에 따라 음성 웹 기술이 빠르게 진보중에 있으며 대표적으로 VoiceXML[2]과 SALT[3]이 있다. 이러한 음성 웹 기술을 이용하여 시각용으로 제작된 웹페이지에서 대화요소를 추출해내고 대화기술로 자동 변환하는 대화기술생성기(Dialog Script Generator)를 개발한 Voice Browser를 위한 Dialog Script Generator의 자동 생성에 대한 연구[4]가 있다. 하지만 이 연구에서는 데스크탑용 웹 페이지인 HTML만을 대상으로 하고 있고 무선 인터넷을 위한 WAP(Wireless Application Protocol) 서비스용 콘텐츠, 예를 들어 WML 기반의 모바일 페이지에 적용하는 기술은 지원되고 있지 않다.

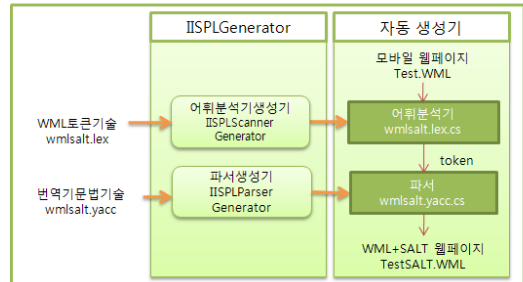
따라서 본 논문에서는 WML로 기술된 무선 인터넷 웹 문서를 음성명령이 가능한 웹 페이지를 위한 WML+SALT언어로 자동변환하기 위한 모바일 음성 웹 페이지의 자동 생성 시스템을 설계하고 구현하고자 한다.

본 논문의 구성은 다음과 같다. 2장에서는 본 논문에서 설계한 자동 생성 시스템의 전체 구성도를 살펴보고, 3장

에서는 WML토큰을 정의한 어휘 분석기 모듈과 4장에서는 WML문서를 WML+SALT로 변환하기 위한 구문지향 번역기 설계를 알아본다. 5장에서는 실험 및 결과를 통해 시스템을 성능을 알아보고 6장에서는 결론 및 향후 연구 방향에 대해 설명하도록 한다.

### 2. 자동생성 시스템 구성

본 논문에서 구현한 모바일 음성 웹 페이지의 자동 생성 시스템의 전체 구성도는 (그림 1)과 같다.



(그림 1) 자동 생성 시스템의 전체 구성도

자동생성기는 하나의 입력과 출력을 가지며 어휘 분석기와 파서로 구성되어 있다. 어휘 분석기(wmlsalt.lex.cs)는 어휘 분석기 생성기(IISPLScannerGenerator)에서 WML 토큰을 기술한 wmlsalt.lex 파일을 입력받아 생성된다. 어휘 분석기는 WML 웹 페이지의 연속적인 문자를 읽어 토큰(token)을 연속적으로 출력한다. 각각의 태그들을 분석하고, 태그와 텍스트를 분리한 후 불필요한 태그들과 주석 등을 제거한다. 또한 필요한 링크 태그요소를 토큰으로 생성하여 파서에게 전달한다. 파서(wmlsalt.yacc.cs)는 파서 생성기(IISPLParserGenerator)에서 변환규칙을 정의한 wmlsalt.yacc 파일을 입력받아 생성된다. 파서는 어휘 분석기에서 받아온 토큰을 이용하여 각 태그에 해당하는 속성들과 문법을 검사한다. Parsing이 완료되면, 변환 규칙에 의해 각각을 대응되는 SALT를 포함한 WML 음성웹페이지로 변환하여 생성한다.

이와 같이 생성되는 과정을 설명하기 위해 <표 1>은 샘플 예제이다. WML 웹 페이지(Test.wml)에서 링크태그는 모바일 환경에서 펜이나 키패드로 선택해야 한다. 링크요소를 SALT를 이용하여 음성으로 제어하기 위해서는 WML+SALT 웹 페이지(Testsalt.wml)에서와 같이 <salt:listen> 엘리먼트로 구성해줘야 한다.

<표 1> WML 웹페이지와 WML+SALT 웹페이지

WML 웹페이지 (Test.wml)	<pre>&lt;wml&gt; &lt;card id="card1"&gt; &lt;p&gt; &lt;a href="#card2"&gt;song&lt;/a&gt;&lt;br/&gt; &lt;anchor&gt;&lt;go href="#card2"/&gt;movie&lt;/anchor&gt;&lt;br/&gt; &lt;do type="accept" label="book"&gt; &lt;go href="#card2" &gt;&lt;/go&gt;&lt;/do&gt; &lt;/p&gt; &lt;/card&gt; &lt;/wml&gt;</pre>
WML+SALT 웹페이지 (Testsalt.wml)	<pre>&lt;wml&gt; &lt;card id="card1"&gt; &lt;p&gt; &lt;a href="#card2"&gt;song&lt;/a&gt;&lt;br/&gt; &lt;anchor&gt;&lt;go href="#card2"/&gt;movie&lt;/anchor&gt;&lt;br/&gt; &lt;do type="accept" label="book"&gt; &lt;go href="#card2" &gt;&lt;/go&gt;&lt;/do&gt; &lt;/p&gt; &lt;salt:listen&gt; &lt;salt:grammar id="card1Reco" xmlns="http://www.w3.org/2001/06/grammar"&gt; &lt;grammar root="root"&gt; &lt;rule naume="card1"&gt; &lt;one-of&gt; &lt;item&gt;song&lt;/item&gt; &lt;item&gt;movie&lt;/item&gt; &lt;item&gt;book&lt;/item&gt; &lt;/one-of&gt; &lt;/rule&gt; &lt;/grammar&gt; &lt;/salt:grammar&gt; &lt;/salt:listen&gt; &lt;/card&gt; &lt;/wml&gt;</pre>

### 3. 어휘 분석기 설계

비주얼용 웹페이지에서는 음성으로 처리될 수 있는 하이퍼링크, 정보를 입력하는 텍스트 박스, 버튼 등이 있다. 이와 같이 사용자로부터 입력 받을 수 있는 명령어들을

음성 요소로 정의할 수 있다. 본 연구에서는 이러한 음성요소들 중에서 하이퍼링크(<A>, <Anchor>, <Do>, <Go>)태그를 음성요소로 정의하기 위해 다음과 같이 각각의 정규표현의 패턴과 토큰, 렉시를 <표 2>와 같이 정의하였다.

<표 2> WML 태그의 정규표현, 토큰, 렉심

정규표현	토큰	렉심의 예
(blank   tab   newline)+	ws	
<	OpenTag	<
>	CloseTag	>
<!(?!)([^\s]*)(>)	COMMENT	<!DOCTYPE wml ....>
card	TCARD	card
p	TP	p
a	TA	a
anchor	TANCHOR	anchor
do	TDO	do
go	TGO	go
...		
id{ws}*={ws}*	AID	id =
href{ws}*={ws}*	AHREF	href =
label{ws}*={ws}*[\^]{N AME}[\^]	SPEECHWORD	label =
[>](NAME)<	SPEECHWORD	> song <, >movie<
{letter}({letter})[0-9]*	NAME	song, movie

어휘 분석기를 생성해 주는 도구로는 본 연구실에서 Lex와 Yacc를 개발한 IISPLGenerator를 이용하였다. IISPLGenerator는 출력을 C#으로 반환하며, 필수적인 사항은 클래스에서 상속받아 쓸 수 있도록 구성되어 있다. (그림 2)는 <표 2>의 WML 토큰들의 정규식 패턴에 따라 정의한 wmlsalt.lex파일의 일부이며, 액션부분은 C#언어로 기술하였다.

```
wmlsalt.lex
{
(Console.WriteLine("TA "+ lexbuf+")); return "TA";
}
p {
(Console.WriteLine("TP "+ lexbuf+")); return "TP";
}
anchor {
(Console.WriteLine("TANCHOR "+ lexbuf+")); return "TANCHOR";
}
do {
(Console.WriteLine("TDO "+ lexbuf+")); return "TDO";
}
go {
(Console.WriteLine("TGO "+ lexbuf+")); return "TGO";
}
id{ws}*={ws}* {
(Console.WriteLine("AID "+ lexbuf+")); return "AID";
}
{>}(NAME)< {
(Console.WriteLine("SPEECHWORD "+ lexbuf+")); return "SPEECHWORD";
}
```

(그림 2) WML 정규표현을 정의한 wmlsalt.lex파일

어휘 분석 생성기는 (그림 2)의 파일을 입력받아 (그림 3)과 같이 어휘 분석기(wmlsalt.lex.cs)파일을 반환하며 토큰과 심볼 테이블(Symbol Table)을 생성한다. 그 결과 37개의 상태와 21개의 토큰을 만들어낸다.

```
case 54:Console.WriteLine("LINK "+lexbuf+"); return "LINK";
case 40:Console.WriteLine("NAME "+lexbuf+"); return "NAME";
case 84:Console.WriteLine("SPEECHWORD "+ lexbuf+"); return "SPEECHWORD";
case 58:Console.WriteLine("TCARD "+ lexbuf+"); return "TCARD";
case 10:Console.WriteLine("TA "+ lexbuf+"); return "TA";
case 11:Console.WriteLine("TP "+ lexbuf+"); return "TP";
case 75:Console.WriteLine("TANCHOR "+ lexbuf+"); return "TANCHOR";
protected override void Install()
{
trans=new int[,]
{
{0,1,<}, {0,2,' '}, {0,2,'\r'}, {0,2,'\n'}, {0,3,'\n'},
{0,4,>}, {0,5,'/'}, {0,6,'w'}, {0,7,'.'}, {0,8,'|'},
{0,9,'c'}, {0,10,'a'}, {0,11,'p'}, {0,12,'d'}, {0,13,'g'},

```

(그림 3) 어휘 분석기(wmlsalt.lex.cs)의 소스파일

#### 4. 파서 설계

파서는 구문지향 번역기이며 문법 구조는 LALR 문법으로 작성되었고, WML 스펙에 정의된 링크 태그와 요소들의 속성 및 속성 값에 대하여 Parsing 할 수 있도록 설계하였다. <표 3>은 WML 태그에 따라 생성규칙을 정의하였다. 생성규칙에서 굵은 글씨는 표현식이며 정자체는 문자열을 의미한다. wml언어는 card태그가 여러 개 올 수 있으며 card태그에 여러 개의 link태그들이 올 수 있는 생성규칙을 정의한 부분이다.

<표 3> WML 태그의 문법 정의

WML 태그	WML 생성규칙
<pre>&lt;wml&gt; &lt;card id="cname"&gt; ...&lt;/card&gt; &lt;card id="cname"&gt; ...&lt;/card&gt; &lt;/wml&gt;</pre>	<b>root-&gt; cards</b>
<pre>&lt;card id="cname"&gt; &lt;do type="type" label="name"&gt; ... &lt;/do&gt; &lt;anchor&gt;name ... &lt;/anchor&gt; &lt;a href="URL"&gt;name&lt;/a&gt; &lt;/card&gt;</pre>	<b>cards-&gt;card cards   card</b> <b>card-&gt;&lt;card id=cname&gt;</b> <b>links-&gt;&lt;link&gt; &lt;/link&gt;</b> <b>links-&gt;link links   link</b> <b>link-&gt;do   a   anchor</b>
<pre>&lt;do type="type" label="name"&gt; ... &lt;/do&gt;</pre>	<b>do-&gt;&lt;do label=SPEECHWORD&gt;</b>
<pre>&lt;a href="URL"&gt;name&lt;/a&gt;</pre>	<b>a-&gt;&lt;a href=url&gt;SPEECHWORD</b> <b>&lt;/a&gt;</b>
<pre>&lt;anchor&gt;name...&lt;/anchor&gt;</pre>	<b>anchor-&gt;&lt;anchor&gt;</b> <b>SPEECHWORD&lt;/anchor&gt;</b>

SALT 스펙에는 음성 인식 및 입력을 위한 <listen> 엘리먼트가 정의되어 있다. 본 연구에서는 링크요소를 음성 인식 대상으로 하기 때문에 <표 4>에서와 같이 WML 생성규칙의 링크요소에서 음성명령 가능한 단어를 추출하여 <listen> 엘리먼트 문법을 구성한다. 음성명령이 가능한 단어는 어휘 분석기에서 SPEECHWORD 토큰으로 반환하며 의미 규칙에 의해 SPEECHWORD를 찾을 때마다 <Grammar>의 <one-of>에 <item>엘리먼트를 구성한다.

<표 4> 번역기 문법 기술

WML 생성규칙	WML+SALT 의미규칙
<b>root-&gt; cards</b>	<b>root.t:=cards.name cards.t</b>
<b>cards-&gt;card cards   card</b> <b>card-&gt;&lt;card id=cname&gt;</b> <b>links-&gt;&lt;link&gt; &lt;/link&gt;</b> <b>links-&gt;link links   link</b> <b>link-&gt;do   a   anchor</b>	<b>cards.name:=card.name cards.name</b> <b>card.name:=&lt;salt:listen&gt;&lt;salt:grammar</b> <b>ar id="name"</b> <b>xmlns="http://www.w3.org.2001/06/gra</b> <b>mmar"&gt;&lt;grammar root=root&gt;&lt;rule</b> <b>name="name"&gt;&lt;one-of&gt;</b> <b>cards.t:=card.t cards.t</b> <b>card.t:=&lt;card id=cname&gt;</b> <b>links.t&lt;card&gt;</b> <b>links.t:=link.t links.t</b> <b>link.t:=do.t   anchor.t   a.t</b>
<b>do-&gt;&lt;do label=SPEECHWORD&gt;</b>	<b>do.t="&lt;item&gt;"speechword"&lt;/item&gt;"</b>
<b>a-&gt;&lt;a href=url&gt;SPEECHWORD</b> <b>&lt;/a&gt;</b>	<b>a.t="&lt;item&gt;"speechword"&lt;/item&gt;"</b>
<b>anchor-&gt;&lt;anchor&gt;</b>	<b>anchor.t="&lt;item&gt;"speechword"&lt;/item</b> <b>&gt;"</b>
<b>SPEECHWORD&lt;/anchor&gt;</b>	
<b>name=&gt;[a-zA-Z0-1]+</b>	

(그림 4)는 번역기 문법을 기술한 wmlsalt.yacc파일의 일부분이다. 하이퍼링크 태그를 만나면 음성 인식으로 사용될 SPEECHWORD 토큰을 nametoken ArrayList에 추가한다.

```
wmlsalt +
link: a | anchor | do ;
a: OpenTag TA Ahref LINK SPEECHWORD / TA CloseTag {nametoken.Add(T5);}
anchor: OpenTag TANCHOR SPEECHWORD / TANCHOR CloseTag {nametoken.Add(T3);}
do: OpenTag TDO ATYPE NAME SPEECHWORD CloseTag OpenTag /TDO CloseTag
{nametoken.Add(T5);}
go: OpenTag TGO Ahref LINK CloseTag OpenTag /TGO
| OpenTag TGO Ahref LINK /
```

(그림 4) 번역기 문법을 정의한 wmlsalt.yacc 파일

파서 생성기는 (그림 4)의 파일을 입력받아 (그림 5)와 같은 파서(wmlsalt.yacc.cs)파일을 반환하여 20개의 문법과 액션테이블을 생성한다.

```
protected override void install()
{
    lexan = new LexicalAnalyzer();
    R.Add("OpenTag","token",0);
    R.Add("CloseTag","token",1);
    R.Add("/","token",2);
    R.Add("TCARD","token",3);
    R.Add("TWML","token",4);
    .....
    P.Add("anchor -> OpenTag TANCHOR SPEECHWORD / TANCHOR CloseTag");
    P.Add("do -> OpenTag TDO ATYPE NAME SPEECHWORD CloseTag OpenTag / TDO CloseTag");
    P.Add("go -> OpenTag TGO Ahref LINK CloseTag OpenTag / TGO");
    P.Add("go -> OpenTag TGO Ahref LINK /");
}
```

(그림 5) 파서(wmlsalt.yacc.cs)의 소스파일

#### 5. 실험 및 결과

##### 5.1 구현환경

본 논문에서 구현한 자동 생성 시스템은 본 연구실에서 개발한 IISPLGenerator를 이용하여 어휘 분석과 파서를 설계하였으며, Visual Studio.NET환경의 C#언어를 이용하여 WML+SALT 웹페이지를 생성하였다.

##### 5.2 실험

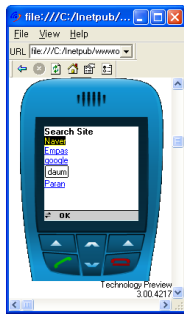
본 연구에서 구현한 모바일 웹 페이지의 자동 생성 시스템은 입력은 WML 웹페이지 파일이고, 출력은 SALT를 포함한 WML 웹 페이지이다.

본 시스템의 성능을 알아보기 위해 다음과 (그림 6)과 같이 WML 모바일 웹 페이지 소스코드를 작성하여 샘플 페이지를 구성하였다. 샘플페이지는 링크요소를 포함하고 있는 소스파일이고, 음성인식 대상의 단어는 Naver, Empas, google, daum, Paran이다.

```
<!DOCTYPE wml PUBLIC "-//WAPFORUM/DTD WML 1.1//EN"
"http://www.wapforum.org/DTD/wml_1.1.xml">
<wml>
<head>
<meta http-equiv="Cache-Control" content="max-age=0"/>
</head>
<card id="card1">
<p><b> Search Site</b></p>
<a href="#card2">Naver</a><br>
<anchor><go href="#card2">Empas</anchor><br>
<anchor><go href="#card2">Google</anchor><br>
<do type="accept" label="daum">
<go href="#card2"></go></do><br>
<a href="#card2">Paran</a></p>
</card>
</wml>
```

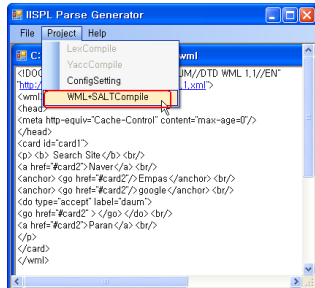
(그림 6) WML 모바일 웹페이지 소스코드

(그림 7)은 소스코드 파일을 Microsoft Mobile Explorer 3.0 에뮬레이터를 통해 실행시킨 결과이다.



(그림 7) WML 웹페이지 에뮬레이터 화면

WML 소스 파일을 WML+SALT 소스를 변환하기 위해 (그림 8)의 자동 생성 시스템에서 wml 파일을 열어 Project메뉴의 WML+SALTCompile 메뉴를 클릭한다. 그 결과 C드라이브에 wmlsalt.wml파일을 생성한다.



(그림 8) 자동 생성 시스템

변환된 wmlsalt.wml 소스는 (그림 9)와 같이 음성명령이 가능한 요소를 추출하여 SALT의 <Listen> 엘리먼트 문법이 추가된 것을 확인할 수 있다.

```
<!DOCTYPE wml PUBLIC "-//WAPFORUM/DTD WML 1.1//EN"
"http://www.wapforum.org/DTD/wml_1.1.xml">
<wml>
<head>
<meta http-equiv="Cache-Control" content="max-age=0"/>
</head>
<card id="card1">
<p>
<a href="#card2">naver</a><br>
<anchor><go href="#card2">empas</anchor><br>
<anchor><go href="#card2">google</anchor><br>
<do type="accept" label="daum">
<a href="#card2">paran</a></do>
</p>
<salt:listen>
<salt:grammar id="name" xmlns="http://www.w3.org/2001/06/grammar">
<grammar root="root"><rule name="name">
<one-of>
<item>naver</item>
<item>empas</item>
<item>google</item>
<item>daum</item>
<item-param/>
<one-of/></grammar></salt:listen>
</card>
</wml>
```

(그림 9) 변환된 WML+SALT 웹 페이지 소스 코드

## 6. 결론

본 연구에서는 모바일 음성 웹 페이지의 자동 생성 시스템을 구현하였다. 자동생성 시스템에서 어휘 분석기는 WML언어로 구성된 웹페이지에서 하이퍼링크 태그들을 토큰으로 추출하였고, 파서에서는 태그들을 번역하고 SALT 변환규칙을 정의하여 문법을 생성하였다.

따라서 개발자는 음성 정보 서비스를 위한 SALT 기반의 콘텐츠를 다시 개발하지 않아도 되며, 이에 따라 저렴한 비용으로 음성 정보 서비스를 구축할 수 있게 되었다. 또한, 모바일 클라이언트 사용자는 키보드뿐만 아니라 음성명령을 통해 웹 브라우저를 제어할 수 있게 되어 입력에 대한 편리성을 증대시켰다.

하지만 현재 모바일 환경에서는 SALT지원해 주는 브라우저가 많지 않다. 따라서 향후에는 모바일이나 PDA에서 SALT를 파싱할 수 있는 브라우저에 대한 연구가 필요하다.

## 참고문헌

- [1] Lettieri P, Srivastava MB, "Adances in wireless teminals", IEEE Personal Communications, V.6 N.1. 6-19.1999
- [2] Scott McGlashan et al., "Vocie Extensible Markup Language(VoiceXML) Version 2.0", W3C Recommendation, 16 March 2004. <http://www.w3.org/TR/voicexml20>
- [3] SALT Forum, "Speech Application Language Tags(SALT 1.0) Specification", 15 July 2002. <http://www.saltforum.org>
- [4] 오지영, "Voice Browser를 위한 Dialog Script Generator의 자동 생성에 대한 연구," 박사논문, 한밭대학교 정보통신대학원, 2008.08
- [5] WAP Forum, "Wireless Markup Language Version 2.0", 11 Sep 2001, <http://www.openmobilealliance.org>
- [6] Alfred V. Aho, "Compilers : Principles, Techniques, and Tools", 사이버출판사, 2006
- [7] 김성국, 장민석, "무선 환경하에서의 음성인식·합성 기반의 대화형 인터페이스 구현", 한국인터넷정보학회, 2006.11,